National Aeronautics and
Space Administration

GRC-CONN-PLAN-0001 B
EFFECTIVE DATE: 06/07/2012

Space Communications and Navigation (SCaN) Testbed Project

National Aeronautics and Space Administration
John H. Glenn Research Center at Lewis Field, OH 44135

SCAN TESTBED PROJECT

SOFTWARE CONFIGURATION MANAGEMENT PLAN

AUTHORIZED by CM when under FORMAL Configuration Control	
Date	**Signature**
06/21/2012	*/s/ James A. Drury*

Distribution:

[ ] NASA (U.S. Gov. Only) [ ] Project Only [X] Government and Contractors

Availability:

[X] Public (No Restriction) [ ] Export Controlled [ ] Confidential/ Commercial [ ] Internal Use Only

PREFACE

National Aeronautics and Space Administration (NASA) is developing an on-orbit, adaptable, Software Defined Radio (SDR)/Space Telecommunications Radio System (STRS)-based testbed facility to conduct a suite of experiments to advance technologies, reduce risk, and enable future mission capabilities on the International Space Station (ISS). The Space Communications and Navigation (SCaN) Testbed Project will provide NASA, industry, other Government agencies, and academic partners the opportunity to develop and field communications, navigation, and networking technologies in the laboratory and space environment based on reconfigurable, software defined radio platforms and the STRS Architecture. The project was previously known as the Communications, Navigation, and Networking reConfigurable Testbed (CoNNeCT). Also included are the required support efforts for Mission Integration and Operations, consisting of a ground system and the Glenn Telescience Support Center (GRC TSC). This document has been prepared in accordance with NASA Glenn's Configuration Management Procedural Requirements GLPR 8040.1 and applies to the SCaN Testbed configuration management activities performed at NASA's Glenn Research Center (GRC). This document is consistent with the requirements of SSP 41170, Configuration Management Requirements, International Space Station, and GLPR 7120.5.30 Space Assurance Requirements (SAR).

This document specifies the Software Configuration Management requirements for the project.

This document was prepared as a Software Configuration Management Plan per requirement SWE-103 of NPR 7150.2, "NASA Software Engineering Requirements". It defines the SCaN Testbed Software configuration items, configuration baselines, tools, processes, and change procedure. The document was prepared using a combination of the Software Configuration Management Plan Template GRC-SW-TPLT-SCM, found at Software@Glenn (http://software.grc.nasa.gov), and the SCaN Testbed standard document template, SCaN Testbed General Plan template (Rev3).dot found in eRoom.

DOCUMENT HISTORY LOG

Status (Preliminary/ Baseline/ Revision/ Canceled)	Document Revision	Effective Date	Description
Baseline	–	02/25/2010	Baseline for System CDR
Revision	A	08/16/2011	Incorporated CONN-CR-0472
Revision	B	06/07/2012	Incorporate CONN-CR-0604 into the document to reflect the current process

SIGNATURE PAGE

Prepared By:

/s/ Tina Grzincic — *06/07/2012*

Tina Gilliotti Grzincic — Date
SCaN Testbed Software Configuration Analyst
ZIN Technologies, Inc.

Concurred By:

/s/ Chris Bodzioney — *06/19/2012*

Christopher Bodzioney — Date
SCaN Testbed Software Product Assurance
ZIN Technologies, Inc.

/s/ Denise Varga — *06/21/2012*

Denise M. Varga — Date
SCaN Testbed Lead Software Engineer
NASA Glenn Research Center

/s/ Charles Hall — *06/11/2012*

Steve Hall — Date
SCaN Testbed Lead Systems Engineer
NASA Glenn Research Center

/s/ Joshua E. Freeh — *06/11/2012*

Josh E. Freeh — Date
SCaN Testbed Chief Engineer
NASA Glenn Research Center

Approved By:

/s/ Diane Malarik — *06/18/2012*

Diane C. Malarik — Date
SCaN Testbed Project Manager
NASA Glenn Research Center

TABLE OF CONTENTS

1.0 INTRODUCTION 1
1.1 Purpose 1
1.2 Scope 1
1.3 SCaN Testbed Computer Software Configuration Items 1
1.4 Relationship to SCaN Testbed Configuration Management Plan 2
2.0 APPLICABLE DOCUMENTS 3
2.1 Reference Documents 3
2.2 Parent Documents 3
2.3 Information Documents 3
2.4 Order of Precedence for Documents 4
3.0 SOFTWARE CONFIGURATION MANAGEMENT (SCM) ORGANIZATION, RESOURCES, AND RESPONSIBILITIES 5
3.1 Organizational Structure and Resources 5
3.2 Responsibilities 6
3.2.1 Software Configuration Management (SCM) - (ZIN) 6
3.2.2 Project Software Engineering Lead - (NASA GRC) 7
3.2.3 Software Engineering Lead (SWEL) - (ZIN) 7
3.2.4 Software Engineers - (ZIN) 7
3.2.5 Software In-Line Product Assurance - (ZIN) 7
3.2.6 Software Oversight Product Assurance - (NASA GRC) 7
3.2.7 Experiment/PI Software Developer 7
3.3 Applicable Policies, Directives and Procedures 8
4.0 SCM ACTIVITIES 9
4.1 Configuration Identification 9
4.1.1 Identifying Configuration Items 9
4.1.1.1 Software Documentation 9
4.1.1.2 Software Executables, Data Files, and Configuration Files 9
4.1.1.3 Support Software 9
4.1.2 Naming Configuration Items 10
4.1.2.1 Version Numbers 13
4.1.2.2 Revision Numbers 13
4.1.3 Management of Software Configurations Items 13
4.2 Configuration Baselines 14
4.2.1 Requirements Baseline 14
4.2.2 Architecture Baseline 14
4.2.3 Design Baseline 14
4.2.4 Development Baseline(s) 14
4.2.5 Production Baseline(s) 15
4.3 Software Configuration Control 15
4.3.1 Level of Control 15
4.3.2 Issue Tracking Tool 15
4.3.3 Software Change Process 16
4.3.4 Software Control Board (SCB) 17
4.4 Software Configuration Status Accounting 18
4.5 Software Configuration Audits 18
4.5.1 Functional Configuration Audit (FCA) 18

4.5.2 Physical Configuration Audit (PCA) 18
4.6 Configuration Reviews 19
4.7 Software Release Management and Delivery Process 19
4.7.1 Software Release Process 19
4.7.2 Release Number Process 21
4.7.3 Version Description Document 22
5.0 SCM SCHEDULE 23
6.0 TOOLS, TECHNIQUES, AND METHODOLOGIES 24
6.1 Planned Backups & Disaster Recovery 24
7.0 CONFIGURATION MANAGEMENT OF ACQUIRED SOFTWARE 25
8.0 RECORDS COLLECTION AND RETENTION 26
APPENDIX A ACRONYMS AND ABBREVIATIONS 27
A.1 Scope 27
A.2 List of Acronyms and Abbreviations 27
APPENDIX B DEFINITIONS 29
B.1 Scope 29
B.2 List of Definitions 29
APPENDIX C NPR 7150.2 COMPLIANCE MATRIX FOR SWE-103 30
C.1 Scope 30
APPENDIX D SCAN TESTBED SOFTWARE CHANGE IMPACT ASSESSMENT (SCIF) FORM 31
D.1 Scope 31

TABLE OF FIGURES

Figure 3-1—SCaN Testbed Project Organizational Structure 6
Figure 4-1—File Revisions 13
Figure 4-2—SCaN Testbed Software Change Process – Approval Cycle 17
Figure 4-3—SCaN Testbed Software Release Process 20

TABLE OF TABLES

Table 1-1—SCaN Testbed CSCI Definition 2
Table 3-1—NPR 7150.2 SCM Requirements Compliance Matrix 8
Table 4-1—SCaN Testbed CSCI Naming Convention 10
Table 4-2—SCaN Testbed PAS Examples 11
Table 4-3—SCaN Testbed Ground Software Examples 12
Table 4-4—Build, Release and Version Examples 21
Table A-1—Acronyms 27
Table B-1—Definitions 29
Table C-1—Traceability from the SCMP Template to SWE-103 of NPR 7150.2 30
Table D-1—SCaN Testbed Software Change Impact Assessment Form (SCIF) 31

1.0 INTRODUCTION

This Software Configuration Management Plan (SCMP) describes the functions, responsibilities, and authority for the accomplishment and implementation of software configuration management (SCM) to be performed during the software life cycle of the SCaN Testbed project. This section provides an overview description of the software project and identifies all of the SCM activities.

1.1 Purpose

This document establishes the SCM activities, tools, methods, and responsibilities for the SCaN Testbed Project. It identifies the required coordination of software configuration management activities with other activities of the project. The plan describes the methodology for establishing configuration identification, change control, status accounting, and performing configuration audits during the development of the software configuration items.

1.2 Scope

This document covers software configuration management for the SCaN Testbed project software, both flight and ground software, including software developed at all partner locations. It is recognized that, due to the distributed nature of the software development team, partner sites may have their own internal configuration management methods, plans, and tools. This document does not cover those internal processes, but covers the partner software once it is delivered to GRC. It is also recognized that some of the software is proprietary in nature to the partner company and that access controls will need to be implemented in order to protect proprietary information.

Software Configuration Management is established and will be used during the life cycle of the SCaN Testbed project. For a complete system overview of the SCaN Testbed project, see Section 1.3 of the Software Requirements Specification, GRC-CONN-REQ-0084.

1.3 SCaN Testbed Computer Software Configuration Items

The following Table 1-1 describes the Computer Software Configuration Items (CSCIs) that make up the SCaN Testbed Software. Note: Other software may have developed-in components such as boot ROMs (Read-Only Memory) and other radio-specific devices and those will be described in the radio documents as opposed to here. The purpose of this CSCI definition is to provide a high-level configuration management organization for the software products that will serve as a structure for setting up the repository. Note also that the Ground Integration Unit software is not considered a separate CSCI, as it will, by definition, run the same software as the flight unit.

Table 1-1—SCaN Testbed CSCI Definition

ID	CSCI Name	Developer	Platform	Purpose
PAS	Payload Avionics Software	GRC/ZIN	Payload Avionics (AiTech S950 Single Board Computer)	Processes commands, data, and telemetry; commands SDRs; routes and stores experimental data; monitors system health; controls heaters; operates gimbals
GD OE	GD Operating Environment	GD	GD SDR (Coldfire Processor)	Process commands; control the configuration of the SDR; execute STRS-compliant Waveforms
GD WF	GD Waveform	GD	GD SDR	Run the S-band transceiver functions for the GD SDR.
JPL OE	JPL Operating Environment	JPL	JPL SDR (Atmel AT697E/Sparc V8 Processors)	Process commands; control the configuration of the SDR; execute STRS-compliant Waveforms
GGT WF	Glenn Goddard TDRS Waveform	GRC/RHD, GFSC	JPL SDR	Run the S-band transceiver functions for the JPL SDR.
GGT WF/OE	GGT Waveform compiled with JPL OE	GRC/RHD	JPL SDR	The combined compiled image containing the JPL OE and the GGT Waveform.
JPL GPS WF	JPL GPS Waveform	JPL	JPL SDR	Run the L-band receiver functions at L1, L2, and L5 for the JPL SDR.
HAR OE	HAR Operating Environment	Harris	Harris SDR (AiTech S950 Single Board Computer)	Process commands; control the configuration of the SDR; execute STRS-compliant Waveforms
HAR WF	HAR Waveform	Harris	HAR SDR	Run the Ka-band transceiver functions for the HAR SDR.
GSW	SCaN Testbed Ground Software	GRC/ZIN	SCaN Testbed Control Center (Intel Processor)	Send commands to payload; receive data from payload; display payload telemetry and data; distribute data to Principal Investigator (PI); store data
TST SW	Test Software	GRC/ZIN	Payload Avionics (AiTech S950 Single Board Computer)	Non-flight software that injects error conditions or provides insight into “white box” software testing.
PIC SW/ WSC SW	SCaN Testbed Ground Software for Principal Investigator Center (PIC) / White Sands Complex	GRC/RHD	PC, 5.0	Software for sourcing and sinking and analyzing data at STCC (PIC). Software for modulating and demodulating signals at White Sands Complex.
GRC ELC Simulator	Ground support equipment (GSE)	GRC/ZIN	Intel PC / Windows XP	Partially emulates the PL MDM & ELC Partially emulates the PayLoad Multiplexer/De-Multiplexer (PL MDM) & ELC

1.4 Relationship to SCaN Testbed Configuration Management Plan

GRC-CONN-PLAN-0002, SCaN Testbed Configuration Management Plan contains Section “3.2 Configuration Management (CM) Organizational Structure, Roles and Responsibilities” which states “The Software Configuration Items and the configuration management methodology applied will be identified in a separate SCaN Testbed Software Configuration Management Plan (GRC-CONN-PLAN-0001).”

This Software Configuration Management Plan discusses the plan for the SCaN Testbed project software. This plan will cover all software code, associated data files, along with other software-related artifacts. All other CM products, i.e. documents, drawings, process plans, etc. are covered in GRC-CONN-PLAN-0002, SCaN Testbed Configuration Management Plan.

2.0 APPLICABLE DOCUMENTS

This section lists the NASA/Government and non-NASA/Government specifications, standards, guidelines, handbooks, or other special publications applicable to the application of this document.

2.1 Reference Documents

Reference documents are those documents that, though not a part of this document, serve to clarify the intent and contents of this document.

Document Number	Reference Document
GLPR-7150.1	GRC Software Engineering Requirements
GRC-7150.9	Software Configuration Management Process
GRC-SW-TPLT-SCMP	Software Configuration Management Plan Template
NPD 2810.1	NASA Information Security Policy
NPR 7150.2	NASA Software Engineering Requirements

2.2 Parent Documents

This section lists the number and title of the latest revision of all documents that are controlling documents to the software project.

Document Number	Reference Document
GRC-CONN-PLAN-0002	SCaN Testbed Configuration Management Plan
GRC-CONN-PLAN-0004	SCaN Testbed Project Plan
GRC-CONN-PLAN-0005	SCaN Testbed Systems Engineering Management Plan
GRC-CONN-PLAN-0024	SCaN Testbed Software Management and Development Plan
GRC-CONN-PLAN-0085	SCaN Testbed Software Assurance Plan
GRC-CONN-REQ-0084	SCaN Testbed Software Requirements Specification

2.3 Information Documents

This section lists the number and title of non-project documents that provide additional information.

Document Number	Reference Document
Book	Software Configuration Management by H. Ronald Berlack
Book	Software Configuration Management – Identification, Accounting, Control, & Management by Steve J. Ayer & Frand S. Patrinostro
CMMI Second Edition - Book	Capability Maturity Model Integrated – Guidelines for Process Integration and Product Improvement
IEEE Std 828-2005	IEEE Standard for Software Configuration Management Plans

2.4 Order of Precedence for Documents

In the event of a conflict between this document and other documents specified herein, the requirements of this document shall apply. In the event of a conflict between this document and higher level documents, the higher level documents shall take precedence over this document.

All documents used, applicable or reference, are to be the approved versions released as of the contract start date. All documents issued after baseline establishment shall be reviewed for impact on scope of work. Nothing in this document supersedes applicable laws and regulations unless a specific exemption has been obtained.

3.0 SOFTWARE CONFIGURATION MANAGEMENT (SCM) ORGANIZATION, RESOURCES, AND RESPONSIBILITIES

This section describes the allocation of responsibilities and authorities for SCM activities and their management to organizations and individuals within the SCaN Testbed Project structure. The SCM activities and processes are discussed in section 4. The SCM software tools, techniques, and methodologies are discussed in Section 6.

3.1 Organizational Structure and Resources

Figure 3-1 shows the SCaN Testbed Project organizational structure. SCM is part of the SCaN Testbed Multi-Center Project Support CM function. SCaN Testbed Project Management is responsible for the decisions, actions, and directions necessary to ensure that the SCMP fulfills the project requirements. ZIN Technologies is responsible for the development of the Flight & Ground software, as well as all CM and SCM support is provided under the Space Development and Operations Contract (SpaceDOC) Delivery Order 128 (DO-128). ZIN Technologies is responsible for implementing and maintaining this SCMP. Resource levels necessary to implement this SCMP will be maintained commensurate with the SCaN Testbed project phase and workload as determined by the NASA SCaN Testbed Project Manager and the ZIN SCaN Testbed Project Lead. The Software team develops the Payload Avionics Software, the Ground Software, and Test Software in accordance with this SCMP. The experiment team members include representatives from NASA GRC, NASA JPL (Pasadena, California), NASA GFSC (Greenbelt, Maryland), GD (Phoenix, Arizona), and Harris (Melbourne, Florida), and may also include universities and private companies. They are responsible for developing experiment/Principal Investigator (PI) provided software – see section 7.0 below for acquired software SCM requirements. Safety and Mission Assurance (S&MA) will perform in-line and oversight software product assurance activities. SCaN Testbed SCM tool administration is provided by NASA GRC.

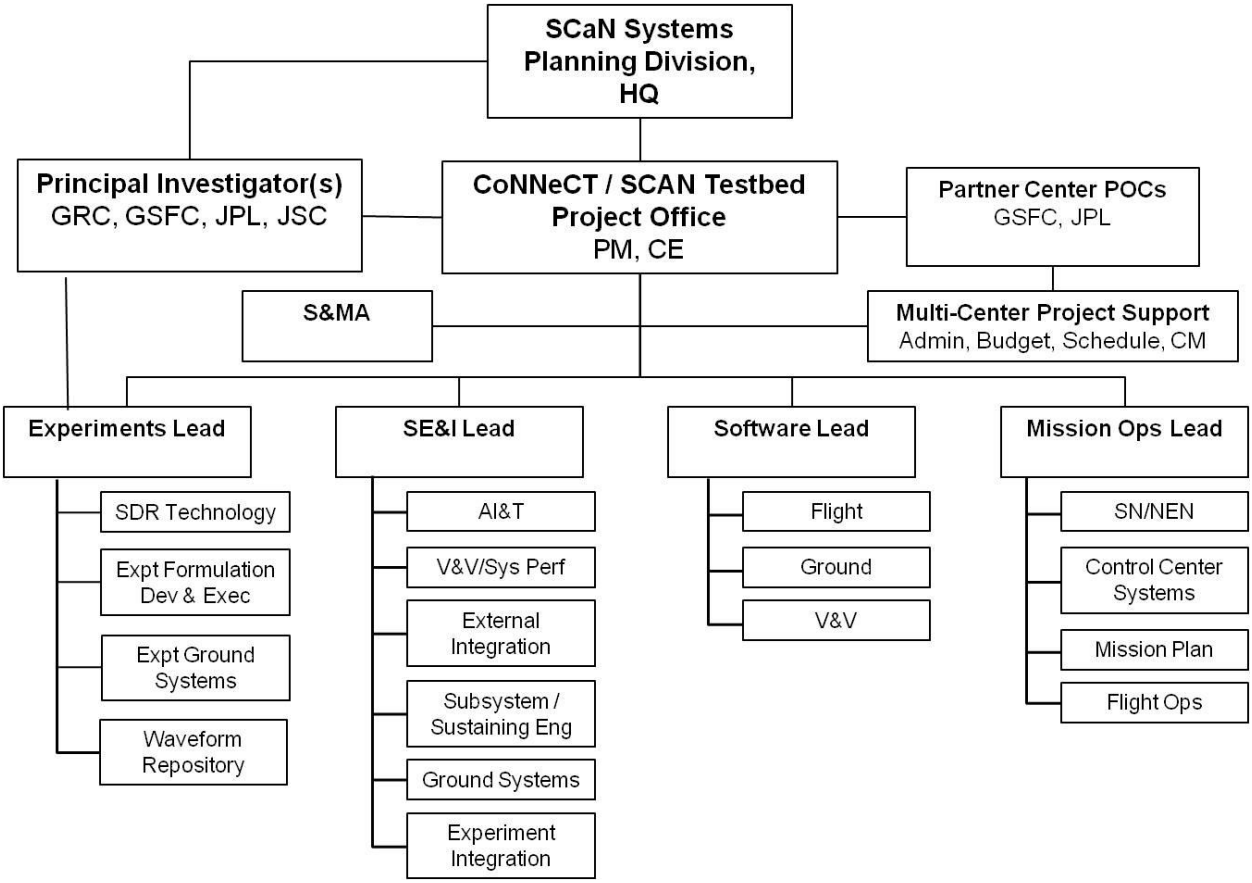

Figure 3-1—SCaN Testbed Project Organizational Structure

3.2 Responsibilities

SCM is responsible for implementing the SCaN Testbed functions detailed in the sections below through direct actions and by coordinating efforts between functional support departments within the project and NASA. The following is a brief overview of the various functions' roles and responsibilities as they pertain to SCM.

3.2.1 Software Configuration Management (SCM) - (ZIN)

- Implement and maintain this SCMP
- Develop, implement, and maintain SCM processes and procedures that support the SCMP activities
- Control all software changes through the Change Requests approval cycle
- Produce all Builds and Releases and associated VDDs
- Perform Configuration Status Accounting
- Control all software baselines and branches

3.2.2 Project Software Engineering Lead - (NASA GRC)

- Establish Software team development priorities
- Serve as a voting member of the Software Control Board (SCB)

3.2.3 Software Engineering Lead (SWEL) - (ZIN)

- Responsible for the integration, testing, verification, and validation of all SCaN Testbed software
- Manage formal problem reporting, Change Requests, and requests for Deviations/Waivers
- Responsible for verifying technical content matches approved changes
- Serve as a voting member of the SCB

3.2.4 Software Engineers - (ZIN)

- Adhere to the SCMP and SCM processes and procedures
- Identify any documentation, process, or Configuration Item viewed as either deficiencies or improvements, and requires review, by generating an Issue (via issue tracking tool), a Nonconformance Report (NCR) or Change Request

3.2.5 Software In-Line Product Assurance - (ZIN)

- Review and approve documentation such as test procedures, process plans, Manufacturing Work Orders, Change Requests, Deviations, and Waivers
- Assure adherence to SCM process and procedures
- Perform Functional Configuration Audit of deliverable software
- Perform Physical Configuration Audit of deliverable software
- Audit the Version Description Document prior to release of deliverable software
- Plan and conduct audits of SCM processes
- Serve as a voting member of the SCB

3.2.6 Software Oversight Product Assurance - (NASA GRC)

- Plan and conduct audits of SCM processes

3.2.7 Experiment/PI Software Developer

- Configuration managed deliverable software per their internal SCM processes.

3.3 Applicable Policies, Directives and Procedures

NPR 7150.2, NASA Software Engineering Requirements levies the following requirements on the project for Class C software. The SCaN Testbed flight and ground software are classified as Class C software. In Table 3-1 NPR 7150.2 SCM Requirements Compliance Matrix depicts how this document shows compliance to each requirement.

Table 3-1—NPR 7150.2 SCM Requirements Compliance Matrix

Number	Description	Requirement	Compliance
4.1.1	The project shall develop a Software Configuration Management Plan that describes the functions, responsibilities, and authority for the implementation of software configuration management for the project.	[SWE-079]	This entire document
4.1.2	The project shall track and evaluate changes to software products.	[SWE-080]	Section 4.3.3 of this document describes the plan to comply with this requirement.
4.1.3	The project shall identify the software configuration items (e.g., software documents, code, data, and scripts) and their versions to be controlled for the project.	[SWE-081]	Section 4.1 of this document.
4.1.4	The project shall establish and implement procedures designating the levels of control each identified configuration item must pass through; the persons or groups with authority to authorize changes and to make changes at each level; and the steps to be followed to request authorization for changes, process change requests; track changes, distribute changes, and maintain past versions.	[SWE-082]	Section 4.3 of this document.
4.1.5	The project shall prepare and maintain records of the configuration status of configuration items.	[SWE-083]	Section 4.4 of this document describes the plan to comply with this requirement.
4.1.6	The project shall ensure that software configuration audits are performed to determine the correct version of the configuration items and verify that they conformed to the documents that define them.	[SWE-084]	Section 4.4 of this document describes the plan to comply with this requirement.
4.1.7	The project shall establish and implement procedures for the storage, handling, delivery, release, and maintenance of deliverable software products.	[SWE-085]	Sections 4.1.1.2, 4.2.3, 4.4, 6.0 & 7.0 of this document describes the plan to comply with this requirement.

4.0 SCM ACTIVITIES

This section describes all SCM activities required to manage the configuration of the software system: software configuration identification, software baselines, software configuration control, software configuration status accounting, software configuration audits and reviews, software release management and delivery.

4.1 Configuration Identification

This section describes the configuration identification process and standards for all Computer Software Configuration Items (CSCIs) in the SCaN Testbed software system.

4.1.1 Identifying Configuration Items

The configuration items shown in Table 4-1 will be developed and managed by the SCaN Testbed project.

4.1.1.1 Software Documentation

Software documentation will be maintained under the project configuration management system per GRC-CONN-PLAN-0002, SCaN Testbed Configuration Management Plan.

4.1.1.2 Software Executables, Data Files, and Configuration Files

There will be one formal SCaN Testbed software repository that is managed by the ZIN Software Configuration Management (SCM) and Software Engineering Lead (SWEL). The Payload Avionics Software (PAS) and any test software will be maintained in the SCaN Testbed software repository during development. Ground Software (GSW) will be maintained in the ZIN configuration management system during development, and all baseline deliveries will be entered into the formal software repository before testing. GRC/GFSC Waveform (GGT WF) software and FPGA code will be maintained by the developers. All baseline deliveries will be entered into formal software repository along with the accompanying Version Description Document (VDD).

4.1.1.3 Support Software

All development tools, operating systems, compilers, and other tools will be configuration managed on a separate branch of the repository. When an upgrade for a tool is installed, the upgraded version will replace the previous version in the repository. This provides a history of which tools were used during each development period. The software configuration management tool Subversion, will not be upgraded unless a bug is uncovered and fixed that impacts development. Minor upgrades will not be implemented, so as to minimize any changes to the CM system. Any changes to tools after the software production baseline has been generated must be approved by the Software Control Board (SCB).

4.1.2 Naming Configuration Items

Names for software documentation items developed at GRC will follow GRC-CONN-PLAN-0002, SCaN Testbed Configuration Management Plan. Names for software documentation and code developed at partner sites will be governed by the authority for that organization, whether GRC-CONN-PLAN-0002, SCaN Testbed Configuration Management Plan or an internal naming scheme, as defined in the partner agreement or the SCaN Testbed Configuration Management Plan. For the purposes of the SCaN Testbed project the term "SCaN Testbed" is to be considered synonymous with "**GRC**."

DEV (Developer) represents the organization responsible for generation of the CSCI (Computer Software Configuration Item). These items include PAS software package, ground software, and PAS deployment files.

CSC (Computer Software Component, representing a computer file name) and Ext (file extension(s)) are generic to standard file naming conventions. CSC names are assigned at the developer's or vendor's discretion; as a result, the DEV-CSCI- naming components are considered understood for files associated with SCaN Testbed software. (See SCaN Testbed Ground Software Examples)

NOTE: Harris, GD, JPL, OE, and WF software are not required to conform to CSCI Naming Conventions listed below.

Configuration item naming convention format is: DEV-CSCI-CSC.Ext as described below:

Table 4-1—SCaN Testbed CSCI Naming Convention

Key	Meaning	Possible Values
DEV	Developer	GRC, GSFC, (SCaN Testbed may be used to represent GRC)
CSCI	Highest level decomposition component name	PAS, GSW, TST, or PAS-xxxx_mm-dd-yyyy-Release(b) (Where xxxx indicates the Subversion number, mm-dd-yyyy represents the month day and year of software build generation, and (b) represents the release branch number if applicable. See examples below.)
CSC	Lowest level decomposition component name	File name; left to choice of developer.
.Ext	File type or extension	.c, .h, .gz, .cpp, .dat, .txt, other valid file extensions, files with no extension or multiple extensions. Example: ffx0-data**.tar.gz.06**

Names for payload avionics software code and executable developed at GRC will follow the following scheme.

Table 4-2—SCaN Testbed PAS Examples

DEV	**CSCI**	**Appended as part of CSCI representative of the release branch**
Connect- Connect = GRC As generated via the SCaN Testbed Build Script.	***PAS-5147_11-30-2011*** Payload Avionics Software- Subversion No.- Date	***-Release5*** Release Branch From Which PAS Was Generated\Assembled.
Connect-PAS-5147_11-30-2011-Release5.0.23		
Connect- Connect = GRC	***PAS-5205_02-02-2012*** Payload Avionics Software- Subversion No.- Date	***-Release5*** Release Branch From Which PAS Was Generated\Assembled.
Connect-PAS-5205_02-02-2012-Release5.0.28		

Note:

CSC and Ext components, (files) of the SCaN Testbed Software builds are included in and delivered in the build release.

DEV GRC (understood)	**CSCI** PAS (understood)	**CSC** File Name: PASOut	**.Ext** File Extensions: **.**tar.gz	**Explanation:** The ***PASOut.tar.gz*** file is documented in the PAS "Build.log" and is a compressed file containing the SCaN Testbed .out executable files for deployment on the Avionics System.

Names for Ground Software code and executables developed at GRC will follow the following scheme.

Table 4-3—SCaN Testbed Ground Software Examples

DEV	**CSCI**	**CSC**	**.Ext**	
GRC (understood)	GSW (understood)	File Name	File Extension (s)	Explanation
Ground Software Executable File				
GRC (understood)	GSW (understood)	CTads	.exe	***CTads.exe*** is an example of a Ground Software Executable File installed on the GSE Computer at C:\Connect\Apps. File is maintained in Subversion at "svn+ssh://(User's ID)@connect-sw.grc.nasa.gov/var/lib/projects/connect-ground/Apps
Ground Software Definitions Files				
GRC (understood)	GSW (understood)	Calibrations	**.**caldef	Files are examples of Definition files installed on the GSE Computer at C:\Connect\Definitions. Files are maintained in Subversion at "svn+ssh://(User's ID)@connect-sw.grc.nasa. gov/var/lib/projects/connect-ground/definitions
		Limits	**.**limdef	
		Enumerations	**.**enmdef	
Ground Software Default Configuration File				
GRC (understood)	GSW (understood)	CoNNeCTTads	**.**cfg	***CoNNeCTTads.cfg*** is an example of a Ground Software default configuration file installed on the GSE Computer at C:\Connect\Apps. File is maintained in Subversion at "svn+ssh://(User's ID)@connect-sw.grc.nasa.gov/var/lib/ projects/connect-ground/Config

4.1.2.1 Version Numbers

The version number is used to identify a specific set of updates to any of the SCaN Testbed software configuration Items (CI). Which are based on the initial release of the SCaN Testbed flight software as well as any software update thereafter. Version numbers are tracked for all software changes for the life of the SCaN Testbed project

4.1.2.2 Revision Numbers

Revision numbers for software documentation will follow GRC-CONN-PLAN-0002, SCaN Testbed Configuration Management Plan.

Software Revision number is an instance of a versioned file that can be recreated throughout the life of the project. Revision numbers for software code and executables will be assigned by the software configuration tool Subversion, which will version control any software item each time a code module is checked in. Subversion has the ability to identify versions of the repository by assigning a unique subversion (SVN) number that is associated with the last commit to the repository, and will also record the date and time of the update, the userid of the developer that is checking in the software, and comments describing the contents of the change. This SVN number is an automatic feature that occurs whenever the Subversion repository performs a check in or a commit. Subversion will attach this unique SVN number to all files in the repository so then each file in the repository will be at its own version at the time of the attachment of this unique SVN number. This SVN number being attached to all files in the repository allows access and retrieval to the entire repository as one configured item.

This is shown in Figure 4-1 File Revisions, the file "FileManager.h" has five (5) revisions: 1.0, 1.1, 1.2, 1.3 and 1.4. The revision number tracks and maintains the revision history of a file throughout the lifecycle of the file along with the SVN number.

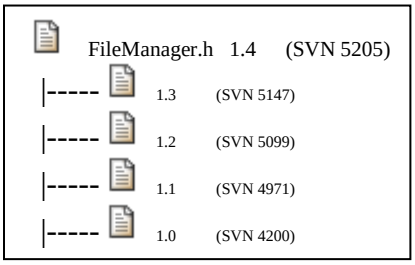

Figure 4-1—File Revisions

4.1.3 Management of Software Configurations Items

A software configuration Subversion repository will be established at GRC in order to store and manage all software code and executables throughout the project. The repository will be accessible via secure VPN internet with a userid and password and access will be given only to project software team members. All proprietary software is controlled through repository is controlled with strict security guidelines. Limited access is controlled to the repository by the project. This repository will be backed up on a regular basis per Section 6.1 of this document.

4.2 Configuration Baselines

This section defines how SCaN Testbed software baselines are established and what each baseline may contain: the number and timing of baselines, items to be controlled in the baseline, the procedures used to create and change the baseline, and the authority required to approve changes to the approved baseline documents. This provides a means of identifying changes and associating them with the affected Configuration Items (CIs), and the associated Configuration Baseline. **Note:** As SDR software is delivered to GRC, it will be placed under version control. However, only the final delivery of SDR software from the partners will be configuration managed with change control. Only the production configuration baseline will contain SDR software.

4.2.1 Requirements Baseline

The Requirements baseline was developed at the time of the software requirements review (SWRR). It includes all documents provided at the SWRR review including; software requirements documents, system requirements, science requirements, concept documents, presentation slides for the review, and all RIDs or comments received as part of the review.

4.2.2 Architecture Baseline

The Architecture baseline was developed after all the software requirements had been flowed down to the lowest level of specification from the SCaN Testbed Software Requirements Specification. It includes any updates to the Requirements Baseline documents, plus all of the lower-level software requirements specifications for each of the software subsystems and CSCIs that evaluate design alternatives; define the subsystem functions, data, and processing logic; and specify technical, hardware, software, operating, security, and interface requirements.

4.2.3 Design Baseline

The Design baseline was developed at the time of the software design review (SWDR) and had included updates to the Architecture Baseline documents, plus the software design documents produced during this phase detailing how the system are implemented and providing the specifications that programmers need to code the modules. **Note:** For SCaN Testbed, the design baseline is not a single event since the flight and ground software maturity is phased over time. The document maturity is shown at the end of the entire phase. There may be added design updates throughout the life of SCaN Testbed project, so this baseline may be changed as required.

4.2.4 Development Baseline(s)

This baseline was developed at the time of software testing and includes updates to the Design Baseline documents, plus program description documents; Unified Modeling Language (UML) descriptions, test plans, and test procedures. All post-ship software is defined per the latest set of Software Requirements governed per GRC-CONN-REQ-0084. These requirements are the ones known at the time of the purposed baseline. Future post-ship software development will produce additional development baselines.

4.2.5 Production Baseline(s)

The Production baseline was developed at the time of the software functional configuration audit/physical configuration audit (FCA/PCA) and includes updates to the Development Baseline documents, all software executables, installation and user guides, test reports, training manuals, version descriptions, and items that were produced during the FCA/PCA. All post-ship software is defined per the latest set of Software Requirements governed per GRC-CONN-REQ-0084, that is known at the time of that purposed baseline. Future post-ship software development will produce additional production baselines.

The following items are identified for the Production Baseline:

- Updates to all other Baselines
- Payload Avionics Software source code and executables, installation and user guides, test reports, training manuals, version descriptions
- Ground Software source code and executables, installation and user guides, test reports, training manuals, version descriptions
- Waveform source code and executables, installation and user guides, test reports, training manuals, version descriptions
- SDR source code and executables, installation and user guides, test reports, training manuals, version descriptions
- FCA/PCA Final Reports

4.3 Software Configuration Control

This section describes the details of the software configuration management steps throughout the Software Life Cycle process.

4.3.1 Level of Control

The official flight release version of the software has been provided to ZIN Technologies. Since the official flight release version has been shipped to Japan, change requests (CRs) are required to modify the software.

4.3.2 Issue Tracking Tool

The project uses an issue change tracking tool, JIRA, to capture all of the project's issues. An issue can be generated by any SCaN Testbed team stakeholder. An issue can be a bug in the software, and anomaly seen during testing, or a request for an improvement or new functionality. Ultimately a change request needs to be generated and approved for any action to be taken on a JIRA request.

4.3.3 Software Change Process

The SCaN Testbed software change process is shown in Figure 4-2. To make a change to the formal baselined software, a change request (CR) must be generated. The change request form is identified in SCaN Testbed Configuration Management Plan, GRC-CONN-PLAN-0002. A Software Change Impact Assessment Form (SCIF) needs to accompany the CR (See Appendix D).

The CR is submitted, after being reviewed and prioritized by the Software Team, the CR is then given approval to be worked by the Software Engineering Lead (SWEL). If there are any disagreements within the Software Team on whether an issue should or should not become a CR, then a Software Control Board (SCB) will be called to review the issue.

Once the CR is approved to be worked a Manufacturing Work Order (MWO) is created. The MWO form is identified in ZIN QMS procedure P09014 "Manufacturing Work Order Procedure". The assigned Developer requests a branch to be created. The SCM creates a branch from the mainline that the developer can use to make the changes. The developer completes all code changes and writes or updates the associated unit tests. And if applicable writes or updates the appropriate CM procedures. Developer works with a tester to verify the change and tests, until all tests have passed. If the software is not verified, the developer goes back to the branch to make the appropriate changes, updates the MWO and continues with testing.

Once the CR has passed branch verification the SWEL reviews the changes associated with the branch. Once the SWEL confirms the code changes, the SCM is given approval to merge the branch to the mainline. If the software change is not confirmed, it goes to the SCB for review. During the merge process the Developer is contacted for support if there are any conflicts. If there are conflicts that need agreements from other associated Developers then the SWEL has finial decision.

After the CR has been merged to the mainline of the repository, the SCM will determine if a build is necessary to be done. A build is done only when the CR change affects more code than just source code. It may affect kernel and data files as well as the source code and would require a complete build to be performed. This build is represented by the subversion SVN number: i.e.: **CoNNeCT-PAS-(SVN)-(DATE)-Release(#),** and once generated will be known as the official Build Number.

Any number of CRs may not require a build to be generated. These CRs may not affect executable files (i.e.: kernel and data files). The SCM will run one of the build scripts, depending upon the type of change the CR falls under. A release will be created whether the CR requires a full build or an incremental delta change set. (See Software Release Management section 4.7 for further details).

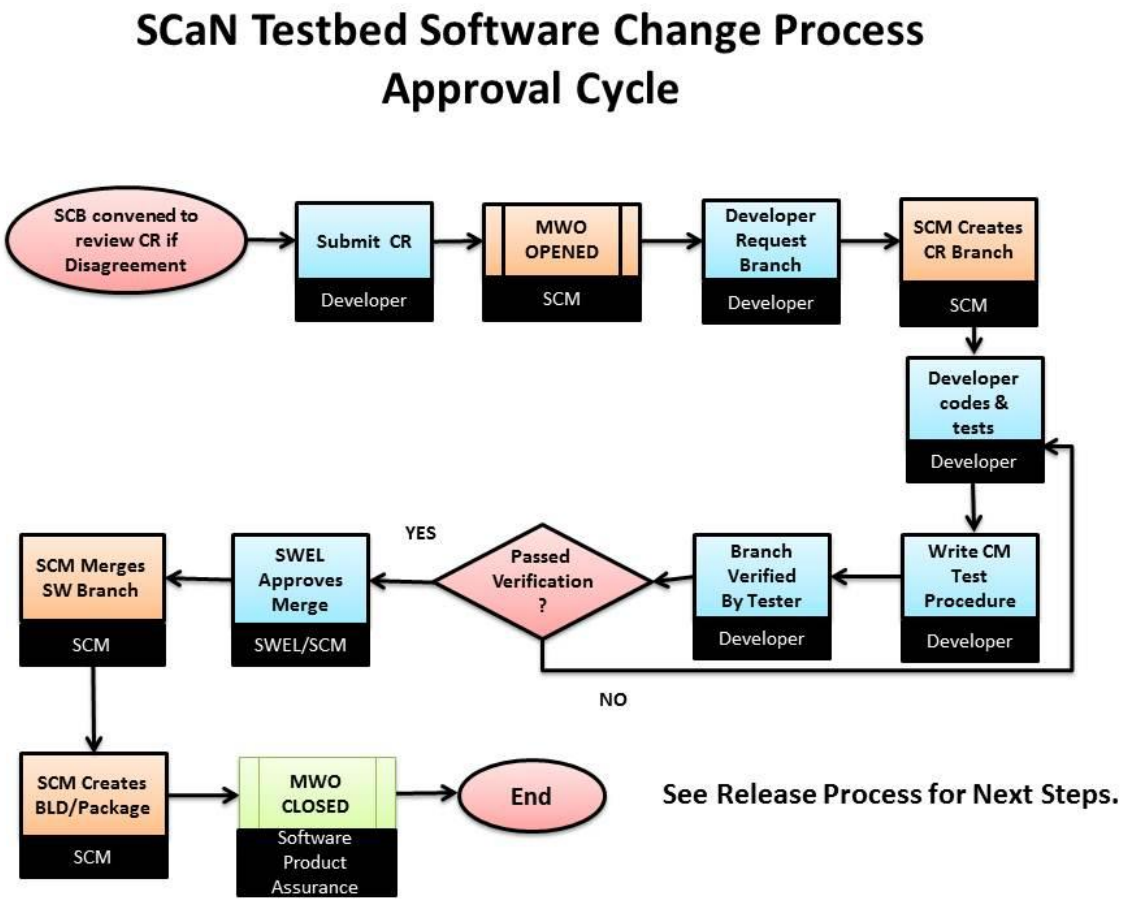

Figure 4-2—SCaN Testbed Software Change Process – Approval Cycle

4.3.4 Software Control Board (SCB)

SCB will only be called to order when required. If there are software team disagreements on submitted issues the SCB will be the deciding factor in determining if an issue is something that needs to be resolved, only if the Software team cannot come to an agreement.

If called, the SCB will be comprised of the following voting members: Project Software Engineering Lead, Software Engineering Lead, Software In-Line Product Assurance representative, Ground Operations representative, and non-voting member, SCM. Developers and other team members may be called in to meetings as required.

4.4 Software Configuration Status Accounting

Configuration status accounting will be used to record and report information needed tomanage the Configuration Items. Will maintain a status record of all items in the baseline(s) and provide traceability of all changes in the system. Configuration status reports will be generated by the ZIN Technologies SCM, prior to a major delivery as defined in the Project Management Plan or when requested per the deliver order. These reports may include the following types of data: configuration item lists, change request reports, number of software modules affected, deviations & waiver status and/or CM metric reports.

4.5 Software Configuration Audits

Configuration audits will be performed to determine if the configuration items accurately reflects the physical and functional characteristics of the SCaN Testbed project. During the initial release of SCaN Testbed software, a Physical Configuration Audit (PCA) and Functional Configuration Audit (FCA) was performed. If software code or executables are changed after the completion of these audits, then new audits must be conducted again. Configuration audit reports will capture the finding from the audit, and list all corrective actions that are required. All subsequent releases of software containing new software loads to be uploaded to the flight system, and the VDD will be updated, and the FCA/PCA reports will be updated to reflect the new software configuration.

4.5.1 Functional Configuration Audit (FCA)

The initial FCA was performed prior to shipping the flight hardware and software to the integration site. The FCA effort verifies that the actual performance of the software agrees with the associated software requirements. And It also verifies that all of the tests for a CI have been completed and that the software performance, based on the test results, meets the specification performance requirements set in the baseline documents. FCA findings may include; traceability verification matrix, listing of all discrepancies, proposed corrective actions, and estimated completion dates.

After the ship date, all new software uploads will undergo an FCA to assess new code and requirements before the upload may be performed. The results of this FCA will be published as an update to the original FCA. NOTE: See Waveform Development Plan for audits to be conducted on new versions of the Waveforms.

4.5.2 Physical Configuration Audit (PCA)

The initial PCA was performed prior to shipping the flight hardware and software to the integration site. The PCA verifies the adequacy, completeness, and accuracy of the software documentation and ensures the correct release version/revision of the software is in the production baseline. The PCA audit may review any of the following items: system deliverables, source code, executables, build reports, release notes, and version description documents. This PCA audit may include the following outputs: audit findings, discrepancies, proposed corrective actions, and estimated completion dates.

After the ship date, all new software uploads will undergo a PCA to assess new and updated documentation, and to verify the new software versions are loaded before the upload may be performed. The results of this PCA will be published as an update to the original PCA.

4.6 Configuration Reviews

SCaN Testbed project reviews will be held as required in support of major deliveries. Software reviews are performed to ensure that CIs have been correctly identified and produced. Any software findings from these reviews will be addressed using the Review Item Discrepancy (RID) forms, and will be handled as in other ZIN Technologies software project reviews.

Software configuration reviews will be performed periodically to verify the correctness of the configuration status accounting information and will enable confirmation to the effectiveness of the SCM process and to identify potential modifications.

4.7 Software Release Management and Delivery Process

The build, release and delivery of software products and documentation will be unique to each development site. All flight software executables will be delivered to GRC in electronic format, either by compact disk or electronic transfer that includes an associated electronic VDD, from a partner repository. The primary objective of the SCM during the release management process is to ensure that the integrity of the production environment is protected and that the correct components are released. Each delivery of source code and executables will be entered into the software repository by the SCaN Testbed SCM.

4.7.1 Software Release Process

The release identification number represents the official release of the software. It contains everything that is part of the build and has a corresponding VDD. Additionally, the Release process can be audited when an official version of the SCaN Testbed software is released. The Release term contains the shortened Release number that is the representation of what has been packaged and will be up loaded to the Flight system.. i.e.: REL#.##.##.

An overview of the Release Process is shown in Figure 4-3. Once the SWEL determines that a release is required, an MWO is created and the SCM generates the official release package, then contacts the QA representative to witness the installation of the release on the GIU.

QA witnesses required V&V testing on the GIU. If verification does not pass, then a NCR is opened which is identified in ZIN QMS procedure P13001 “Procedure For The Control Of Nonconforming Product”. If verification passes, then the REL#.##.## is released. A release notice is sent out. And the VDD is updated (See section 4.7.3) This software will be installed on other hardware as requested.

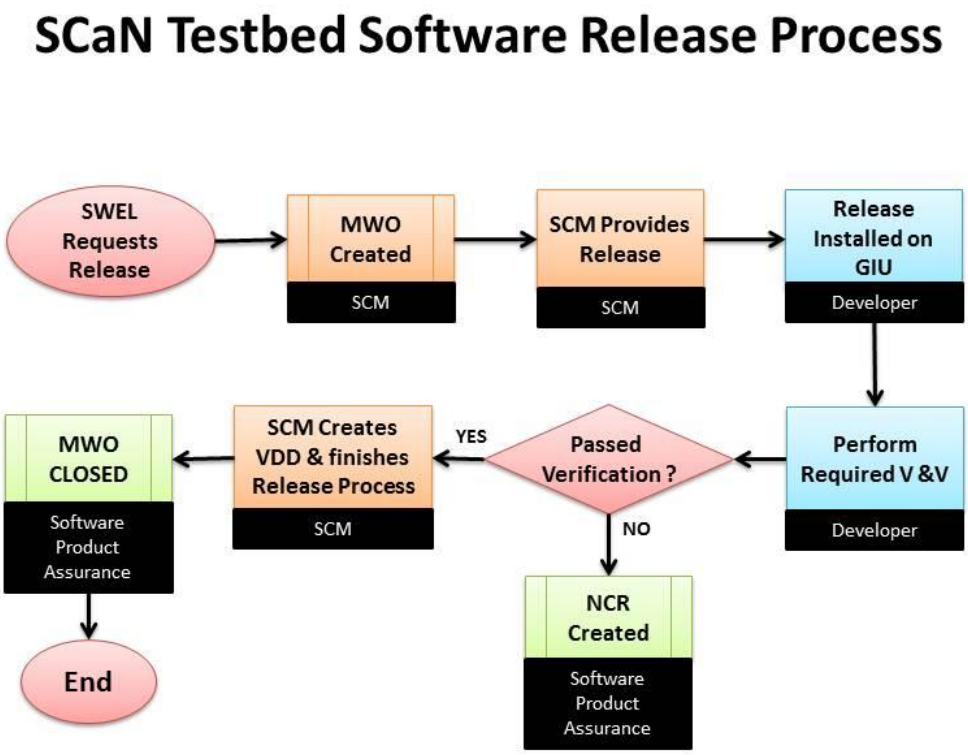

Figure 4-3—SCaN Testbed Software Release Process

4.7.2 Release Number Process

The Release number reflects [**Major.Minor.Daily**] from either an incremental delta change set or full build. These terms are defined as follows:

- **Major** designation of software, like the value "1" in a version designation of 1.0".
- **Minor** designation of software, like the value "2" in a typical version designation of "1.2".
- **Daily** designation of software, like an ALPHA or a Numeric and tracks only change sets (deltas). This is represented by the term VERSION. The VERSION field represents the daily incremental change to the repository mainline. And is represented as the third position, like the value "3" in a typical version designation of "1.2.3". i.e.: ##**.##.##**

The following Table 4-4 shows an example of the various types of BUILDS, VERSIONS and RELEASES associated with the the SCaN Testbed project.

Table 4-4—Build, Release and Version Examples

Change Number	Build Number	Version	Files Affected	Release	Change Set	Comments
888, 877, 866	CoNNeCT-PAS-5749-11302011-Release5	5.30.0	All files built and included in this Release package	REL5.30.0	All files	This is the initial Release package. So it contains a full set of all files.
899, 898	CoNNeCT-PAS-5749-11302011-Release5	5.30.1	Foo.c out.a cmd.d txt.f …	none	12 files – non-build	Did not warrant a release. Was not up loaded to Flight.
944	CoNNeCT-PAS-5749-11302011-Release5	5.30.2	xxx.out sss.out aaa.out	none	3 files non-build	" " " ".
955	CoNNeCT-PAS-5999-12132011-Release5	5.30.3	cmd.out	REL5.30.3	Includes this 1 file from JIRA 955, as well as only the changed files that are associated with the following JIRAs: 944, 899, & 898.	Uses Individual file script. This is an upload of just the changed files since REL5.30.0 release.
1022	CoNNeCT-PAS-6734-01142012-Release5	5.31.0	All files built and included in this Release package	REL5.31.0	Includes all changed files from JIRA 1022, as well as any affected files that were changed with this new build	Uses Build to Build script & packages up load only the changes. This is a bigger package than the individual file script.

4.7.3 Version Description Document

Each SCaN Testbed PAS release will have all associated changes, listed per CR, identified in the SCaN Testbed PAS VDD. Each revision of the SCaN Testbed PAS VDD will always map back to the associated SCaN Testbed PAS release number. For any subsequent releases that are depended upon future changes, these new releases will be associated through a revision change to the SCaN Testbed PAS VDD. Approved deviations and waivers must be documented in the VDD for each release.

5.0 SCM SCHEDULE

See the SCaN Testbed Project Schedule for project milestones and dates. Interim releases of software for hardware interface and subsystem testing will be created based on hardware delivery schedules and will be identified and maintained on the software project schedule.

6.0 TOOLS, TECHNIQUES, AND METHODOLOGIES

The tool chosen to implement SCM on the project is the Subversion version control software. This is an open source version control system and can be found at http://subversion.tigris.org/.

For deliveries of software for hardware, subsystem, and system testing, an associated SubVersion Number (SVN) is generated in Subversion to preserve the configuration for that test. The avionics software development team will be working on individual branches, based on approved Changes. Each branch will have the latest mainline version of the SCaN Testbed software. This will allow developers to work on certain aspects of the code without being impacted by changes being implemented by other team members.

The mainline branch will be the latest version of the SCaN Testbed Flight Software, and will be tightly controlled. All development will be done on individual branches and after being successfully tested, verified and approved by the SCB, will be merged back to the mainline by the SCM representative.

Other tools may interface to the repository, such as development environments (Workbench for VxWorks, Microsoft Visual Studio), development tools (Fisheye, Crucible), and bug tracking tool (Jira). The repository resides on a Linux platform. A Windows tool, TortoiseSVN, will be used to create local repositories on developers' machines. TortoiseSVN can be found at http://tortoisesvn.tigris.org/.

6.1 Planned Backups & Disaster Recovery

The System Administrator for the server that will house the SCaN Testbed software repository will perform backup procedures. Daily incremental backups will be done during the height of the critical software development. This backup will be kept in off-line storage at a separate site at GRC from the software repository. A backup server is run that syncs with the main servers repositories on an hourly basis. All SQL data is mirrored in real time. At all times, two previous backups will be kept to protect against data loss should a backup become damaged or unusable. Restoration of data should be available in a 24 hours period upon detection of data loss.

7.0 CONFIGURATION MANAGEMENT OF ACQUIRED SOFTWARE

Code delivered with the software defined radios will be configuration managed in the GRC repository only after delivery to GRC. These software defined radio codes will have an associated VDD supplied to GRC at the time of delivery. Changes are not made to this part of the repository unless a problem is found in testing and the developer implements a fix. This repository is kept in order to verify that the latest code base is installed on the flight hardware correctly. It also allows investigation of problems that may be found by the GRC team during software integration and testing. Due to the proprietary nature of some of this code, access will be controlled to only those with authority to have access to the code, per agreement with the partner. During software testing at GRC, any problems found in acquired software will be recorded on an NCR.

8.0 RECORDS COLLECTION AND RETENTION

The software configuration management records will be maintained under the Space Development and Operations Contract (SpaceDOC) Delivery Order 128 (DO-128) controlled by ZIN Technologies.

APPENDIX A ACRONYMS AND ABBREVIATIONS

A.1 Scope

This appendix lists the acronyms and abbreviations used in this document.

A.2 List of Acronyms and Abbreviations

Table A-1—Acronyms

ANSI	American National Standards Institute
CRB	Configuration Review Board
STCC	SCaN Testbed Control Center
CI	Configuration Item
CM	Configuration Management
CMMI	Capability Maturity Model Integrated
CoNNeCT	Communications, Navigation, and Networking reConfigurable Testbed
CPAR	Corrective and Preventive Action Report
CR	Change Request
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
ELC	EXpedite the PRocessing of Experiments to Space Station (EXPRESS) Logistics Carrier
FCA	Functional Configuration Audit
GD	General Dynamics
GGT	GRC GSFC TDRS (Waveform)
GIU	Ground Integration Unit
GPS	Global Positioning System
GRC	Glenn Research Center
GSFC	Goddard Space Flight Center
GSW	Ground Software
HAR	Harris
IEEE	Institute of Electrical and Electronics Engineers
ISS	International Space Station
JPL	Jet Propulsion Lab
MWO	Manufacturing Work Order
NASA	National Aeronautics and Space Administration
NCR	Nonconformance Report
OE	Operating Environment
PAS	Payload Avionics Software
PCA	Physical Configuration Audit
PIC	Principal Investigator Center
RID	Review Item Discrepancy
RF	Radio Frequency
ROM	Read-Only Memory
QMS	Quality Management System
SAR	Space Assurance and Requirements
SCaN	Space Communications and Navigation
SCB	Software Control Board

SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDR	Software Defined Radios
SM&A	Safety and Mission Assurance
STD	Standard
STRS	Space Telecommunications Radio System
SVN	SubVersion Number
SWEL	Software Lead
SWRR	Software Requirements Review
TDRSS	Tracking and Data Relay Satellite System
TRL	Technology Readiness Level
TSC	Telescience Support Center
TST	Test
VDD	Version Description Document
WF	Waveform
WSC	White Sands Complex
ZIN	ZIN Technologies, Inc.

APPENDIX B DEFINITIONS

B.1 Scope

The definitions used in this section conform to the definitions found in the ANSI/IEEE Std 729-1983, *IEEE Standard Glossary of Software Engineering Terminology.* See specifically: configuration management, configuration identification, configuration control, configuration status accounting, configuration audit, configuration control board, baseline, promotion, release, version, and revision.

B.2 List of Definitions

The glossary contains an alphabetized list of definitions for special terms used in the document, i.e., terms used in a sense that differs from or is more specific than the common usage for such terms.

Table B-1—Definitions

Term	Definition
Baseline	A controlled version of all items required to release a particular product.
Build	Is a complete build of the source code or a subset of the code. This could include an incremental or a full build.
Change Request	A formally submitted artifact that is used to track all stakeholder requests (including new features, requests, defects, changed requirements, etc.) along with related status information throughout the project life cycle.
CM Repository	A controlled database that holds all versions of the lowest level components of the software project.
Configuration	The functional and physical characteristics of hardware, firmware, software, or documentation which is controlled.
Configuration Audit	CM Audit is defined as either an FCA or a PCA review, depending upon the level of control required, as well as the schedule set for the project.
Configuration Identification	CM identification includes the selection of CIs, determines the types of configuration documentation required for each CI, the issuance of numbers and identifiers to the CIs, including technical documentation that defines the CI configuration, the release of CIs and associated documents.
Configuration Status Accounting	The recording and reporting of information required to manage configuration effectively. This includes a record of approved documentation and identification numbers, along with all proposed, approved and incorporated change requests.
FCA	Functional Configuration Audit – is to verify that a CIs actual performance agrees with its software requirements as stated in its SRS and IRS
JIRA	Software Issue and Change Tracking Tool
PCA	Physical Configuration Audit – is to determine if the design and product specifications and referenced documents and physical software load represent the software that was coded and tested for a specified project.
Release	Represents the official release of the software. It contains everything that is part of the build and has a corresponding VDD.
Version	Version represents the major, minor and daily incremental change to the repository mainline

APPENDIX C NPR 7150.2 COMPLIANCE MATRIX FOR SWE-103

C.1 Scope

Requirement SWE-103 of NPR 7150.2 "NASA Software Engineering Requirements" levies a set of requirements that must be met by a project's Software Configuration Management Plan. The table below lists the NPR 7150.2 requirements for a SMP and the corresponding section(s) of this SCaN Testbed Software Configuration Management Plan that address those requirements.

Table C-1—Traceability from the SCMP Template to SWE-103 of NPR 7150.2

SCMP Section(s)	SWE-103 Requirement
3.0 SCM Organization and Resources	a. The project organization(s) within which Software Configuration Management is to apply.
3.1 Organizational Structure	b. Responsibilities of the software configuration management organization.
2.0 Applicable Documents	c. References to the software configuration management policies and directives that apply to the project.
4.0 SCM Activities 4.1 Configuration Identification 4.3 Software Configuration Control 4.4 Configuration Status Accounting 4.5 & 4.6 Software Configuration Audits & Configuration Reviews	d. All functions and tasks required to manage the configuration of the software, including: configuration identification configuration control status accounting configuration audits and reviews
4.0 SCM Activities 5.0 SCM Schedules	e. Schedule information, which establishes the sequence and coordination for the identified activities and for all events affecting the Plan's implementation.
6.0 Tools, Techniques, and Methodologies	f. Resource information, which identifies the software tools, techniques, and equipment necessary for the implementation of the activities.
6.1 Planned Backup & Recovery	g. Plan maintenance information, which identifies the activities and responsibilities necessary to ensure continued planning during the life cycle of the project.
4.7 Release Management and Delivery Process	h. Release management and delivery.

APPENDIX D SCAN TESTBED SOFTWARE CHANGE IMPACT ASSESSMENT (SCIF) FORM

D.1 Scope

The SCaN Testbed SCIF form is required to be filled out for any change that is submitted into the system. The SCIF forms will be attached to the corresponding CR.

Table D-1—SCaN Testbed Software Change Impact Assessment Form (SCIF)

Submitter Name		**JIRA Number**	
Date Requested		**Date Required**	
JIRA Description			
Safety Critical?	**_YES**	**NO**	

Change Detail
Likelihood of Occurrence
Consequences if not Implemented (including Science Impacts and Requirements not met)
Available Operational Workarounds

Impact Analysis		
Work products to be modified	**Affected (√)**	**Version number**
1. avionics.out		
2. connect.out		
3. hwif.out		
4. telemetry.out		
5. connectlib.out		
6. MainPowerControl.out		
7. RedundantPowerControl.out		
8. HarrisPowerControl.out		
9. TwtaPowerControl.out		
10. antennaManager.out		
11. connectbasic.out		
12. kernel – vxWorks		
13. kernel – vxWorksAlt		
14. kernel – vxWorksBasic.out		
15. CTADS		
16. CESDB		
17. TReK		
18. ELC Simulator Software		
19. Other (specify)		
State foreseen impact of the suggested changes.		

Risk of Implementing Change		
Verifications that will be affected	**Affected (√)**	**Requirements number(s)**
1. RT Validation		
2. KSC ELC Simulator C&DH Testing		
3. Flight System Software V&V		
4. GIU Software V&V		
5. Ground Software Verification		
6. Other (specify)		
State the impact to the verifications that have already been performed for each requirement and required regression testing.		

Schedule Impact & Scope of Work			
Deliverables Description	**Effort Hours**	**Date Required**	**Resources Required**
1.			
2.			
3.			
Based on the impact, state the projected timing consequences of making the requested change. List the tasks and work effort in person-hours.			

SCB Decision	
☐ Approved	☐ Rejected
☐ Approved with modifications	☐ Deferred
Justifications	